Aturan Praktikum:

- Keterlambatan maksimal 30 menit
- Hasil praktikum, selesai atau tidak selesai ditanggung mahasiswa
- Nilai praktikum adalah nilai perorangan, sebagai tambahan nilai test atau quiz jika nilai test atau quiz kurang mencukupi
- Mahasiswa login ke komputer
- Persiapan dan penjelasan praktikum 10 menit
- Praktikum dilakukan selama 70 menit terdiri dari praktikum sesuai modul dan sebuah soal praktikum
- Hasil praktikum di-zip dan dikumpulkan
- Sisa waktu 20 menit digunakan untuk tes kecil hasil praktikum, saat tes tampilan komputer adalah background desktop dan mahasiswa tidak diperkenankan menggunakan komputer dan melihat modul praktikum
- Mahasiswa logout dari komputer
- Praktikum selesai

Common Object Request Broker Architecture (CORBA)
(Minggu 8 – Praktikum 2)

1. Praktikum

a. Persiapan

- Membuat direktori kerja dengan nama SI319-P8-2-Kelas-NIM misalnya SI319-P8-2-A-23507024
- Di dalam direktori di atas, buat direktori Calculator untuk menyimpan file-file yang akan dibuat.

b. Corba Calculator dengan Java

i. IDL

Membuat file dengan nama Calculator.idl pada direktori Calculator sebagai berikut:

```
module CalculatorApp
{
  interface Calculator
  {
    double count(in string args);
    oneway void shutdown();
  };
};
```

ii. Server

Membuat file CalculatorServer.java

```
import CalculatorApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.PortableServer.POA;

import java.util.Properties;

class CalculatorImpl extends CalculatorPOA {
  private ORB orb;

  public void setORB(ORB orb_val) {
    orb = orb_val;
  }

  // implementasi metode sayHello()
  public double count(String args) {
      String arr[] = args.split("\\s+");
      double result = 0;
      try{
            int i = 0;
            double operan1 = 0;
            double operan2 = 0;
            while(i < arr.length){
                  if(i == 0){
                        operan1 = Double.parseDouble(arr[i]);
                  }else{
                        // tanda aritmetika
                        if(i != 1){
                              operan1 = result;
                        }
                        if(arr[i].equalsIgnoreCase("+")){
                              // operasi tambah
                              i++;

                              operan2 = Double.parseDouble(arr[i]);
                              result = operan1 + operan2;
                        }else if(arr[i].equalsIgnoreCase("-")){
                              // operasi kurang
                              i++;

                              operan2 = Double.parseDouble(arr[i]);
                              result = operan1 - operan2;
                        }else if(arr[i].equalsIgnoreCase("*")){
                              // operasi kali
                              i++;

                              operan2 = Double.parseDouble(arr[i]);
                              result = operan1 * operan2;
                        }else if(arr[i].equalsIgnoreCase("/")){
                              // operasi bagi
                              i++;

                              operan2 = Double.parseDouble(arr[i]);
                              result = operan1 / operan2;
                        }
```

```
                }
                i++;
            }
      }catch(Exception e){
            e.printStackTrace();
      }
    return result;
  }

  // implementasi shutdown()
  public void shutdown() {
    orb.shutdown(false);
  }
}


public class CalculatorServer {

  public static void main(String args[]) {
    try{
      // membuat dan menginisialisasi ORB
      ORB orb = ORB.init(args, null);

      // membuat referensi ke rootpoa dan mengaktifkan POAManager
      POA                          rootpoa                          =
POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
      rootpoa.the_POAManager().activate();

      // membuat servant dan meregisternya ke ORB
      CalculatorImpl calculatorImpl = new CalculatorImpl();
      calculatorImpl.setORB(orb);

      // mengambil objek referensi dari servant
      org.omg.CORBA.Object                      ref                     =
rootpoa.servant_to_reference(calculatorImpl);
      Calculator href = CalculatorHelper.narrow(ref);

      // mengambil root naming context
      // NameService memanggil layananan penamaan
      org.omg.CORBA.Object objRef =
          orb.resolve_initial_references("NameService");
      //  Menggunakan  NamingContextExt  yang  merupakan  bagian  dari
Interoperable
      // Spesifikasi Naming Service (INS).
      NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

      // bind objek yang diacu dengan menggunakan penamaan
      String name = "Calculator";
      NameComponent path[] = ncRef.to_name( name );
      ncRef.rebind(path, href);

      System.out.println("CalculatorServer ready and waiting ...");

      // menunggu panggilan klien
      orb.run();
    }
```

```
      catch (Exception e) {
        System.err.println("ERROR: " + e);
        e.printStackTrace(System.out);
      }

      System.out.println("CalculatorServer Exiting ...");

  }
}
```

iii. Client

Membuat CalculatorClient.java

```
import CalculatorApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import java.io.*;

public class CalculatorClient
{
  static Calculator calculatorImpl;

  public static void main(String args[])
    {
      try{
        // membuat dan menginisialisasi ORB
            ORB orb = ORB.init(args, null);

        // mengambil root naming context
        org.omg.CORBA.Object objRef =
          orb.resolve_initial_references("NameService");
        /* menggunakan NamingContextExt yang merupakan bagian dari
                Interoperable naming Service */
        NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

        // mengacu Object yang diacu penamaan
        String name = "Calculator";

            // mengkhususkan objek ncRef.resolve_str(name)
            calculatorImpl =
                 CalculatorHelper.narrow(ncRef.resolve_str(name));

            System.out.println("Nama : [isi dengan nama]");
            System.out.println("NIM  : [isi dengan NIM]");
            System.out.println("Masukkan hitungan yang ingin dihitung,"
                  + " misal 2 + 3 - 1 : ");

            // membaca masukan
            BufferedReader is =
               new BufferedReader(new InputStreamReader(System.in));
            String hitung =  is.readLine();
            System.out.println();
            if(!hitung.equals("")){
```

```
                    System.out.println("Hasil perhitungan dari : "
                       + hitung
                       + " adalah " + calculatorImpl.count(hitung));
              }else{
                    System.out.println("Masukan tidak boleh kosong");
              }

              // mematikan server
          calculatorImpl.shutdown();

        } catch (Exception e) {
            System.out.println("ERROR : " + e) ;
              e.printStackTrace(System.out);
          }
      }

}
```

iv. Menjalankan aplikasi

- Buka Command prompt, masuk ke direktori dimana file .idl disimpan
- Jalankan kompiler IDL-to-Java (idlj) pada file .idl untuk membuat stub dan skeletons dengan perintah (jika perintah tidak dikenali, set path ke jdk):

  ```
  idlj -fall  Calculator.idl
  ```

- Ketik perintah:

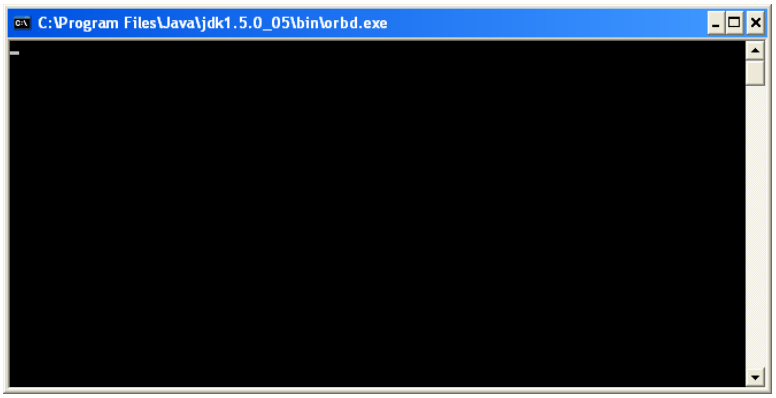
  ```
  javac -Xlint *.java CalculatorApp/*.java
  ```

  akan muncul beberapa warning, abaikan. Jika muncul error perbaiki.
- Ketik perintah:

  ```
  start orbd -ORBInitialPort 1050
  ```

  yang berarti bahwa port 1050 dipilih untuk naming naming service, port dapat diganti port lain jika port 1050 telah terpakai, jika port tidak dituliskan pada perintah maka port standar yang digunakan adalah 900, jika muncul error, coba ganti port.

  hingga muncul jendela berikut:



- ketik perintah

  ```
  start java -cp . CalculatorServer -ORBInitialPort 1050 -
  ORBInitialHost localhost
  ```

  hingga muncul jendela commad prompt dengan tulisan "`CalculatorServer ready and waiting`"
- ketik perintah

  ```
  java -cp . CalculatorClient -ORBInitialPort 1050 -ORBInitialHost
  localhost
  ```

  catatan:
  jika server atau client dijalankan pada mesin yang berbeda maka perintah yang harus dijalankan adalah:

  ```
  java –cp . [CalculatorServer/CalculatorClient] -ORBInitialHost
  namerserverhost -ORBInitialPort 1050
  ```

- Matikan ORB dengan perintah (pada Command prompt orbd) dengan menekan tombol ctrl+C

v. Laporan

Laporan terdiri dari:

- Print screen hasil di sisi client
- Kesimpulan hasil eksekusi program
- Bandingkan kode program di sisi client dan server dengan praktikum Minggu 4 Praktikum 1 (Hello CORBA) bagian mana saja yang berubah